UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/020,033	12/06/2001	Attila D. Banki	PM 2000.063	8954

34477 7590 03/01/2010

EXXONMOBIL UPSTREAM RESEARCH COMPANY
P.O. Box 2189
(CORP-URC-SW 359)
Houston, TX 77252-2189

EXAMINER	
PROCTOR, JASON SCOTT	

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
03/01/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

RECORD OF ORAL HEARING

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte ATTILA BANKI, and STEPHEN C. NETEMEYER

Appeal 2009-008229
Application 10/020,033
Technology Center 2100

Oral Hearing Held: February 4, 2010

Before JAMES D. THOMAS, LANCE LEONARD BARRY, and STEPHEN C. SIU, *Administrative Patent Judges.*

APPEARANCES:

ON BEHALF OF THE APPELLANT:

MATTHEW T. SHANLEY, ESQUIRE
DAVID JENSEN, ESQUIRE
Exxon Mobil Upstream Research Company
P.O. Box 2189
CORP-URC SW 359
Houston, Texas 77252-2189

The above-entitled matter came on for hearing on Thursday, February 4, 2010, commencing at 10:01 a.m., at the U.S. Patent and Trademark Office, 600 Dulany Street, Alexandria, Virginia, before Paula Lowery, Notary Public.

THE CLERK: Good morning. Calendar Number 25, Mr. Shanley.

MR. SHANLEY: Good morning. My name is Matthew Shanley. I'm the attorney representing the assignee, Exxon Mobil Upstream Research Company. I'm here today to talk about in particular one point of distinction that we believe distinguishes this application over the prior art record.

What I'd like to direct your attention to are a couple of things. The first one is Figure 1 of our own application, which I think is pretty representative of at least the field we're talking about here.

In addition, there was an additional figure that we produced in the context of the Appeal Brief to try to explain further to the Examiner, after seeing some of his comments in the Examiner's answer. That's on page 6 of the Reply Brief. It also shows up in the Appeal Brief.

JUDGE THOMAS: Has the Examiner entered that as a formal drawing in the application?

MR. SHANLEY: As a formal drawing? No, it was not.

JUDGE THOMAS: Okay.

MR. SHANLEY: Instead it was a graphic representation of the statements that were already made in both the response at the final and also in the Appeal Brief. But there should be no additional matter in there other than what was already argued.

JUDGE THOMAS: You may proceed.

MR. SHANLEY: With respect to the prior art, I think it would be Figure 1-2, which shows up on page 1-8 of the reference, that's been referred to by both the Examiner and us as the Real Time Workshop.

JUDGE THOMAS: We only have two screens here, Counsel. What are the things you would like us to focus on first? We can only look at two pictures at a time.

MR. SHANLEY: Two pictures at a time?

JUDGE THOMAS: In the record, yes.

MR. SHANLEY: Sure. Quickly, look at Figure 1 of the present, that'll give you a high level, and then with Claim 1 in the back of your mind here -- if you can pull that up --

JUDGE THOMAS: Yeah, that's what we usually have.

MR. SHANLEY: With Figure 1 we have a computer system for simulating a physical system. The example that we use through the specification and what we're really discussing in detail in many of the dependent claims is a reservoir simulator.

If you see in Figure 1, this is what a three-dimensional model of the subsurface might look like. The model itself are the cells that you see here. What you're looking at there on Figure 1 is actually the first layer. Depicted underneath it are subsequent layers that would have additional details.

But based on these cells you can get an idea of how fluids flow. You can also trace chemical properties to really understand the physics of how things behave beneath the subsurface.

In addition to that, what you also see is another aspect here. These are shown generally with bold lines with the letters T, N and also W -- dotted lines where you see that.

That would be what we'd typically refer to in our field as facilities. These are the physical facilities such as pumps, compressors, wells, things that are interacting with this subsurface region in order to either pump out fluids, or in some cases inject fluids to try to stimulate the reservoir to enhance production.

That's an example that's far more than what you actually see in Claim 1, but it's representative of some of the details.

With that said, I think we can probably turn to page 6 of the Reply Brief, the figure that sort of follows more the process. In addition, we have Figure 1-2, which is in the prior art.

In Claim 1 we have a processor, and we have a memory coupled to the processor. We specifically recite object-oriented software because it is very suitable for this sort of application. The points of contention between us and the Examiner really come down to the sub-elements here: A, B, C.

What I'd really like to focus on today is sub-element A. We discussed sub-elements B, C, and D in the Briefs, and I think those arguments can stand as presented. With respect to A, I'd like to walk you through some of our arguments or positions there.

Most importantly, we have this software that's configured to provide a logic interface to dynamically construct logic, to customize simulation or transform phenomena through a model of the physical system.

If you recall the reservoir, the physical system is the reservoir. The logic interface is the ability for, in our case, our end users -- the facilities engineer or reservoir engineer. This is the person who in the course of a simulation of an actual reservoir, this may take days, weeks, months, years, depending on the complexity of the model to run.

It allows an engineer to come in and say, well, if I had the ability to alter the rules which relate to particular pumps, compressors, et cetera, I could go in and manipulate how this reservoir simulator might respond. So dynamically you can go in and alter the rules that apply to this actual model.

JUDGE BARRY: So this is a user interface? What we used to call a man-machine interface, right?

MR. SHANLEY: That's right.

JUDGE BARRY: Okay.

MR. SHANLEY: Exactly right, and one of the figures here actually talks about a GUI --

JUDGE BARRY: Oh, there's a GUI? Okay.

MR. SHANLEY: One of the benefits of this by doing object-oriented principles, you're allowed to have a flow chart that essentially links all of these elements because individual pumps and compressors are ultimately connected to various other pieces of equipment. So it's important to see how they relate in the overall hierarchy of equipment.

So a user can come in and manipulate the logic. This is not getting into the actual data --

JUDGE BARRY: Right.

MR. SHANLEY: -- or the simulator itself; but it's the ability to control so as you go through the individual time steps in the reservoir, the logic introduced dynamically allows the ability for the user to manipulate on the fly what's actually taking place in the reservoir.

So for the first five years of operation the reservoir may behave one way. For example, you may pump oil and gas out of the ground without having to stimulate it at all because natural pressures do it. As you get more and more

mature and drag more and more of these hydrocarbons up, you may need to inject things. So maybe the original model presented that it would be ten years before we thought we would have to inject fluids. Perhaps we learn that problems are coming up much sooner. This would allow the logic to be manipulated and to go ahead and tune how this thing might respond.

I think one of the major points of contention between ourselves and the Examiner has been -- really it comes down to a few words. It's providing a logic interface to dynamically construct logic to customize simulation of the transport phenomena through a model of a physical system.

It does get into element B here, this is what we particularly emphasized in the Reply Brief. Converting this logic into corresponding object-oriented code during the simulation and without intervention of the user.

By that, what it means is, if you go to page 6 in the Reply Brief, that's the figure that's sort of a flow chart that we think is pretty helpful in walking you through what's happening there on the claim elements.

For example, we have at the very top of the chart -- it sort of forms a branching network. On the right-hand side of the page you see a bubble that says executable main simulation system. This would be your typical reservoir simulator that's dealing with the physical system. The physical system could include some facilities, but it's the actual overall plumbing and how the earth is.

On the left-hand side is this logic interface. What you see here, and what we have bracketed here to demonstrate, is this notion of during the simulation -- the word single run is used in there, but there's explicit support for that in the specification -- the word single run; but we'd essentially argue that "during the simulation" and "single run" are the analogous terms.

What it allows you to do is once the user has constructed this logic is to allow this logic to be absorbed into the main simulation system on the fly as it's actually simulating, automatically generate the code -- again during the simulation -- which you see down here where it merges; and then produce a fully-integrated system that has the former reservoir simulation system in it with now the additional logic you decided to dynamically add at that point.
In contrast, if we head over to Figures 1-2, which is on page 1-8 of the Real Time Workshop reference, I think that is very representative of what's going on. The Real Time Workshop in and of itself is a module within a larger suite of applications. The Real Time Workshop is a way to generate code for things that are produced in either MATLab or simulated up above.
Our intention is not to argue the merits or details of the individual modules, but it's important to note that because of the way this is set up as a modular system that has different aspects, and the way it's specifically described, every example of how you apply MATLab, Simulink, and Real Time Workshop involves -- in this case it's more of a programmer that's the end user on MATLab and Simulink.
As they develop their actual software, they need to run through the simulation. They need to check it. As you see in the flow chart on Figure 1-2, this is depicted by -- let's say MATLab was used as a problem solver or something to analyze the actual physical system and help create a model. Simulink is what is used to simulate this model, and as you're I'm sure familiar with, there are things such as iterating to get to a solution. To see how the system behaves that they've attempted to describe.
Most importantly, this triangle in the center that says "are the results okay"? What the user is now determining is is my model adequate enough? Has it

been simulated to the point that it works? If not, it goes back up into the same system here.

If it is okay, the user jumping in -- they have the ability to invoke Real Time Workshop, which is to generate the code. Without that user interaction, the system essentially doesn't work.

The Examiner has a secondary reference, which is essentially C++ programming, which is certainly an example of object-oriented programming. No dispute there. I think it's important to note that even if one were to try to go back and try to edit this to become an object-oriented programming system, there still is no teaching or suggestion that you would have this sort of temporal aspect.

That you would go ahead and try to do this dynamically during the course of the simulation, as opposed to post-simulation, which is explicitly described in every example that we can see in --

JUDGE THOMAS: Counselor, can I sort of summarize what I think you're saying?

MR. SHANLEY: Absolutely.

JUDGE THOMAS: It seems like Clauses B and C recite the negative limitation without intervention of the simulator user.

MR. SHANLEY: Yes.

JUDGE THOMAS: That's the focus of your argument, is it not? The combination from the best perspective of the Examiner is that it requires for actual usability the user's intervention.

MR. SHANLEY: I think well said. In fact, I think I'm done with the majority of the presentation here, and that's how I would probably best summarize it.

The only thing I might add is if you look at the Examiner's Answer where he's particularly spoken to this point, we could -- as far as the exact page --

JUDGE THOMAS: Yes, would you --

MR. SHANLEY: The recitation that the Examiner points to for that exact distinction and the way you've framed it shows up on -- bear with me one second. Page 18 of the Examiner's Answer. There's a bold headline that says "Response to Argument 4."

If you go down, we have this without intervention of the simulation user, and it points to the fact that the Real Time Workshop talks about automatically building programs.

I think that's correct except for the fact that it misses the second part of it which is when it's taking place. So it's not only without user intervention, it's that we're doing it during the course of the simulation itself is when you're allowing to merge this new logic.

The Examiner is absolutely correct. This doesn't talk about automatically building things, but automatic doesn't necessarily include that it's doing an automatic throughout the entire process. There's clear points of intervention by the user.

JUDGE BARRY: Can you focus on independent Claim 20? I believe we understand the negative limitation approach with respect to independent Claim 1, but Claim 20 doesn't seem to have the same features. Am I understanding that correctly?

MR. SHANLEY: We've chosen to argue them together. Let me just walk through and see if there's a particular point maybe I'm missing.

We have the analogous limitations here fall under Step B in this method, and then we have I or number 1 and then Part 2 that we have.

JUDGE BARRY: Okay. So B and the automatic feature.
MR. SHANLEY: Right, so we have automatic merging of logic into code --
JUDGE BARRY: Oh, okay. So it's an umbrella of negative limitation.
MR. SHANLEY: Right.
JUDGE BARRY: I see.
MR. SHANLEY: This has more specificity in that it recites a reservoir simulator user as opposed to the word reservoir didn't show up in Claim 1. So I believe the Examiner's position on that was that it wasn't referenced elsewhere in the claims and wasn't necessarily given patentable weight by the Examiner. I think it still comes back to the same core argument in Claim 1 as far as the distinction here.
Unless you have additional questions, that's what I planned on walking through.
JUDGE THOMAS: It appears the panel doesn't, so we thank you.
Whereupon, the proceedings at 10:15 a.m. were concluded.